CLAIM AMENDMENT

Please amend the claims in accordance with the following listing:

Listing of Claims

Claims 1-60 (Canceled).

61. (New): A method comprising the steps of:

receiving, at a second cache, a cache request for an uncacheable object in a current state, the cache request being generated by a first cache in response to a client request;

requesting, at the second cache, information regarding the uncacheable object in the current state from a server;

obtaining, at the second cache, a tag value associated with the uncacheable object in the current state;

determining whether the first cache is likely to have the uncacheable object in the current state, the step of determining being performed in response to the step of obtaining.

62. (New): A method as in claim 61, further comprising:

sending the uncacheable object in the current state from the second cache to the first cache if the step of determining indicates that the first cache is unlikely to have the uncacheable object in the current state.

63. (New): A method as in claim 62, wherein the step of determining is performed at the second cache.

64. (New): A method as in claim 63, wherein the uncacheable object is not sent to the first cache in response to the cache request if (1) the step of determining indicates that the first cache is likely to have the uncacheable object in the current state, and (2) the first cache has the uncacheable object in the current state.

65. (New): A method as in claim 64, further comprising sending to the first cache an indication that the uncacheable object is not sent to the first cache in response to the cache request if the step of determining indicates that the first cache is likely to have the uncacheable object in the current state.

66. (New): A method as in claim 65, further comprising the step of sending the uncacheable object from the second cache to the first cache if (1) the step of determining does not indicate that the first cache is likely to have the uncacheable object in the current state, or (2) the first cache does not have the uncacheable object in the current state.

67. (New): A method as in claim 66, wherein the step of obtaining comprises:

computing, at the second cache, an MD5 signature of the uncacheable object in the current state.

68. (New): A method as in claim 65, wherein the step of determining comprises:

maintaining, at the second cache, a bitmap associated with the uncacheable object, the bitmap including a bit indicating whether the first cache has a copy of the uncacheable object.

69. (New): A method as in claim 65, wherein:

the step of determining comprises maintaining, at the second cache, a bitmap associated with the uncacheable object, the bitmap including a first bit indicating whether the first cache has a copy of the uncacheable object, and a second bit indicating whether a third cache has a copy of the uncacheable object; and

the step of requesting comprises requesting, at the second cache, the uncacheable object from the third cache when the second bit indicates that the third cache has a copy of the uncacheable object and the first bit indicates that the first cache does not have a copy of the uncacheable object.

70. (New): A method as in claim 65, wherein the step of requesting comprises:

requesting, at the second cache, information regarding the uncacheable object in the current state from a server.

71. (New): A method as in claim 65, wherein the step of requesting comprises:

requesting, at the second cache, the uncacheable object in the current state from a server.

72. (New): A memory storing information including instructions executable by a processor of a root cache, the instructions, when executed by the processor, causing the root cache to perform the following steps:

receiving a cache request for an uncacheable object in a current state, the cache request being generated by a first leaf cache in response to a client request;

requesting information regarding the uncacheable object in the current state from a server;

obtaining a tag value associated with the uncacheable object in the current state;

determining whether the first leaf cache is likely to have the uncacheable object in the current state, the step of determining being performed in response to the step of obtaining.

73. (New): A memory as in claim 72, wherein the instructions, when executed by the processor, further cause the root cache to send the uncacheable object in the current state from the root cache to the first leaf cache if the step of determining indicates that the first leaf cache is unlikely to have the uncacheable object in the current state.

74. (New): A memory as in claim 73, wherein the root cache does not send the uncacheable object to the first leaf cache in response to the cache request if (1) the step of determining indicates that the first leaf cache is likely to have the uncacheable object in the current state, and (2) the first leaf cache has the uncacheable object in the current state.

75. (New): A memory as in claim 74, wherein the instructions, when executed by the processor, further cause the root cache to send to the first leaf cache an indication that the uncacheable object is not sent to the first leaf cache in response to the cache request if the step of determining indicates that the first leaf cache is likely to have the uncacheable object in the current state.

76. (New): A memory as in claim 75, wherein the instructions, when executed by the processor, further cause the root cache to send the uncacheable object from the root cache to the first leaf cache if (1) the step of determining does not indicate that the first leaf cache is likely to have the uncacheable object in the current state, or (2) the first leaf cache does not have the uncacheable object in the current state.

77. (New): A memory as in claim 75, wherein the step of obtaining comprises:

computing an MD5 signature of the uncacheable object in the current state.

78. (New): A memory as in claim 75, wherein the step of determining comprises:

maintaining a bitmap associated with the uncacheable object, the bitmap including a bit indicating whether the first leaf cache has a copy of the uncacheable object.

79. (New): A memory as in claim 75, wherein:

the step of determining comprises maintaining a bitmap associated with the uncacheable object, the bitmap including a first bit indicating whether the first leaf cache has a copy of the

uncacheable object, and a second bit indicating whether a second leaf cache has a copy of the uncacheable object; and

the step of requesting comprises requesting the uncacheable object from the second leaf cache when the second bit indicates that the second leaf cache has a copy of the uncacheable object and the first bit indicates that the first leaf cache does not have a copy of the uncacheable object.

80. (New): A memory as in claim 75, wherein the step of requesting comprises:

requesting information regarding the uncacheable object in the current state from a server.

81. (New): A memory as in claim 75, wherein the step of requesting comprises:

requesting the uncacheable object in the current state from a server.